Multidimensional Trellis Coded Phase Modulation Using a Multilevel Concatenation Approach—Part II: Codes for the AWGN and Fading Channels

Sandeep Rajpal, *Member, IEEE*, Do Jun Rhee, *Member, IEEE*, and Shu Lin, *Fellow, IEEE*

***Abstract*— In this paper, we will use the construction technique proposed in [1] to construct multidimensional trellis coded modulation (TCM) codes for both the additive white Gaussian noise (AWGN) and the fading channels. Analytical performance bounds and simulation results show that these codes perform very well and achieve significant coding gains over uncoded reference modulation systems. In addition, the proposed technique can be used to construct codes which have a performance/decoding complexity advantage over the codes listed in literature.**

***Index Terms*— AWGN channel, fading channel, multidimensional MPSK TCM codes.**

I. Introduction

As WAS POINTED out in [1], for modulation codes over the additive white Gaussian noise (AWGN) channel, the main parameter of interest is the minimum squared Euclidean distance between the transmitted code sequences and the number of nearest neighbors. Details on the above parameters are available in [2] and [3], and as such, we will not reiterate these design considerations here. The aforementioned design considerations will be the basis of construction of the modulation codes for the AWGN channel in this paper.

If the channel is changed to a fading channel, most codes designed for the AWGN channel no longer perform well, simply because the design parameters of a modulation code which need to be optimized for the fading channel are different from that for the AWGN channel. For the fading channel, we shall consider two scenarios. For the first case, we shall consider the Rayleigh-fading channel with slow fading, coherent detection, no channel state information, independent symbol fading and minimum squared Euclidean distance as the decoding metric. These assumptions have been considered so as to enable us to compare our codes with the ones listed in literature. Examples 3 and 4 construct codes for this scenario. For the second case, we consider the MSAT channel with light shadowing. Example 5 constructs a code for this case.

We would like to add that the code construction technique is universal and is by no means restricted by the aforementioned assumptions. For the fading channels in general, the error performance of a code primarily depends on its minimum symbol distance, the minimum product distance and path multiplicity. It depends on the minimum squared Euclidean distance to a lesser extent. Detailed discussion on these parameters of interest is given in [4] and [5]. As such, we will not reiterate these design considerations here. The dominant parameter of interest is, however, the minimum symbol distance, and as such we will concentrate on optimizing this parameter, when we construct codes for the fading channel.

This paper is organized as follows: In Section II of this paper, we will derive general analytical bounds on the performance of the modulation codes using the multistage decoding techniques proposed in Part I of this paper. In Section III, we will construct examples using the proposed technique and compare them with the codes listed in literature.

II. Performance Analysis

In this section, we will derive a general expression for the bit-error probability (BEP) of the multidimensional trellis coded modulation (TCM) codes decoded using the multistage technique proposed in [1, Section V].

For $1 \leq i \leq q$, let X_i be a random variable, where the value of X_i denotes the number of bit errors at the ith decoding stage at a particular time instant t. Hence, $0 \leq X_i \leq k_i$. Then, the BEP of the multidimensional TCM code, denoted $P_b(e)$, is

$$P_b(e) = \mathbf{E}\left(\sum_{i=1}^{q} X_i\right) \Big/ \sum_{i=1}^{q} k_i = (\mathbf{E}(X_1) + \mathbf{E}(X_2) + \cdots + \mathbf{E}(X_q)) \Big/ \sum_{i=1}^{q} k_i \tag{2.1}$$

where $\mathbf{E}(\cdot)$ denotes the expectation operator. For $2 \leq i \leq q$, $\mathbf{E}(X_i)$ can be broken up into two terms, the first one being the expected number of errors at the ith stage assuming that the

Paper approved by D. Divsalar, the Editor for Coding Theory and Applications of the IEEE Communications Society. Manuscript received March 3, 1995; revised May 3, 1996. This work was supported by the NSF under Grant NCR-9115400 and under Grant NCR-9415374, and NASA under Grant NAG 5-931. This paper was presented in part at the 1993 International Symposium on Communications, Taiwan, R.O.C., December 7–10, 1993.

S. Rajpal is with Rockwell International Corporation, Multimedia Communications Division, Newport Beach, CA 92660 USA.

D. J. Rhee is with LSI Logic Corporation, Milpitas, CA 95035 USA.

S. Lin is with the Department of Electrical Engineering, University of Hawaii at Manoa, Honolulu, HI 96822 USA.

Publisher Item Identifier S 0090-6778(97)01406-2.

previous $i-1$ stages of decoding are correct and the second one being the expected number of errors at the ith stage due to erroneous decoding at either one of the previous $i-1$ stages of decoding, i.e., the error propagation term. Hence,

$$\mathbf{E}(X_i) \le (\mathbf{E}(X_i)|_{\text{error propagation}} \cdot p_{E_i}) + \mathbf{E}(X_i)|_{\text{ith stage error}} \tag{2.2}$$

where $\mathbf{E}(X_i)|_{\text{error propagation}}$ denotes the error propagation term, p_{E_i} denotes the probability of error propagation from the previous stages and $\mathbf{E}(X_i)|_{\text{ith stage error}}$ denotes the term due to erroneous decoding at the i-stage, assuming that the previous $i-1$ stages of decoding are correct. Hence, (2.1) can be rewritten in the following form:

$$P_b(e) \le \left(\sum_{i=2}^{q} (\mathbf{E}(X_i)|_{\text{error propagation}} \cdot p_{E_i}) + \sum_{i=1}^{q} (\mathbf{E}(X_i)|_{\text{ith stage error}}) \right) \Big/ \sum_{i=1}^{q} k_i. \tag{2.3}$$

Except for a few specific cases, it is not possible to obtain a general expression for the expected number of bits in error due to error propagation. The expected number of bits in error due to error propagation depend on both the choice of the inner codes as well as the outer codes, as will be shown in the examples to be discussed later in this paper. As such, we will therefore derive a general expression for the rest of the terms in (2.3).

Let $\mathbf{V}$ be the transmitted code sequence. Using [1, eq. (3.12)] $\mathbf{V}$ can be written in the form $\lambda(\phi_1(\mathbf{v_1}) + \phi_2(\mathbf{v_2}) + \cdots + \phi_q(\mathbf{v_q}))$, where $\mathbf{v_i}$ for $1 \le i \le q$ denotes a code sequence in the convolutional code at the ith stage, C_i.

For $1 \le i \le (q-1)$, let us consider the term $\mathbf{E}(X_i)|_{\text{ith stage error}}$. Recall from [1, Section V] that at the ith stage of decoding, we form the trellis $\lambda(\tilde{C}_i)$, where a code sequence in $\lambda(\tilde{C}_i)$ is of the form $\lambda(\phi_1(\hat{\mathbf{v}}_1) + \phi_2(\hat{\mathbf{v}}_2) + \cdots + \phi_{i-1}(\hat{\mathbf{v}}_{i-1}) + \phi_i(\mathbf{u_i}) + \omega_i)$, where $\mathbf{u_i}$ is a code sequence in the convolutional code at the ith level, C_i, ω_i is a sequence of points from Ω_i, and for $1 \le j \le (i-1)$, $\hat{\mathbf{v}}_\mathbf{j}$ denotes the estimate of $\mathbf{v_j}$. Since we are considering the term $\mathbf{E}(X_i)|_{\text{ith stage error}}$, $\hat{\mathbf{v}}_j = \mathbf{v_j}$ for $1 \le j \le (i-1)$. Also, since C_i is a linear code, the code sequence $\mathbf{u_i}$ can be written in the form $\mathbf{u_i} = \mathbf{v_i} + \mathbf{e}$, where $\mathbf{e}$ is code sequence in C_i. As such, any code sequence in $\lambda(\tilde{C}_i)$ can be rewritten in the form $\hat{\mathbf{V}} = \lambda(\phi_1(\mathbf{v_1}) + \phi_2(\mathbf{v_2}) + \cdots + \phi_{i-1}(\mathbf{v}_{i-1}) + \phi_i(\mathbf{v_i} + \mathbf{e}) + \omega_i)$. Say, that the decoder at the ith stage of decoding decodes the code sequence associated with the convolutional code to be $\mathbf{v_i} + \mathbf{e}$, and let the probability that the event occurs be $p_\mathbf{e}$. The exact expressions for $p_\mathbf{e}$ can be found in [2] and [3] for the AWGN channel and in [4] for the Rayleigh-fading channel. Let $I_\mathbf{e}$ denote the number of nonzero information bits associated with the sequence $\mathbf{e}$. Then the expected number of bits in error (per decoding time instant) due to the sequence $\mathbf{e}$ is $I_\mathbf{e} \cdot p_\mathbf{e}$. Since $\mathbf{e}$ is any arbitrary code sequence in the convolutional code C_i, the total number of bits in error at the ith stage, $\mathbf{E}(X_i)|_{\text{ith stage error}}$ is obtained by considering all the possible code sequences and adding up all the $I_\mathbf{e} \cdot p_\mathbf{e}$ terms, i.e.,

$$\mathbf{E}(X_i)|_{\text{ith stage error}} \le \sum_{\mathbf{e} \in C_i} I_\mathbf{e} \cdot p_\mathbf{e} \tag{2.4}$$

where C_i denotes the set of all the code sequences in the convolutional code, C_i.

Special Case—AWGN Channel: For the results derived above, let us consider the special case when the channel is AWGN. Let $\mathbf{V}$ be the transmitted code sequence and let $\hat{\mathbf{V}}$ be the decoded code sequence. Both these sequences have the form as given earlier. Let $D_\mathbf{e}^2$ denote the *minimum* squared Euclidean distance between $\mathbf{V}$ and $\hat{\mathbf{V}}$. Since $\mathbf{v_j}$ for $1 \le j \le (i-1)$ is arbitrary, $D_\mathbf{e}^2$ has been taken to be the *minimum* over all possible transmitted code sequences for a fixed $\mathbf{e}$. This is the worst case scenario, and as such the minimum squared Euclidean distance $D_\mathbf{e}^2$ gives us an upper bound on the performance of the code. Also, let $N_\mathbf{e}$ be the number of codewords at a squared Euclidean distance of $D_\mathbf{e}^2$ from $\mathbf{V}$. The probability that $\mathbf{V}$ is decoded incorrectly depends upon both $D_\mathbf{e}^2$ as well as $N_\mathbf{e}$ [3]. The code sequences $\mathbf{v_i}$ and $\mathbf{e}$ can be written in the general form $\mathbf{v_i} = (v_{i,1}, v_{i,2}, \cdots, v_{i,p}, \cdots)$, and $\mathbf{e} = (e_1, e_2, \ldots, e_p, \cdots)$, where $v_{i,p}$ and e_p for $1 \le p \le \infty$ denotes the output sequence (n_i bits) of $\mathbf{v_i}$ and $\mathbf{e}$, respectively, at the pth time instant. The minimum squared Euclidean distance between $\hat{\mathbf{V}}$ and $\mathbf{V}$ at the pth time instant depends only on e_p and let this squared Euclidean distance be denoted by $D_{e_p}^2$. Also, let N_{e_p} be the corresponding number of nearest neighbors [3]. Then, $D_\mathbf{e}^2 = \sum_{p=1}^{\infty} D_{e_p}^2$ and $N_\mathbf{e} = \prod_{p=1}^{\infty} N_{e_p}$. $D_\mathbf{e}^2$ and $N_\mathbf{e}$ can be evaluated using the technique proposed in [3].

$\mathbf{E}(X_q)|_{q\text{th stage error}}$ depends on whether the qth level of encoding uses a convolutional code or is left uncoded. If a convolutional code is used at the qth level, then the expressions for $\mathbf{E}(X_q)|_{q\text{th stage error}}$ are the same as those derived above. However, if the qth level is left uncoded then $\mathbf{E}(X_q)|_{q\text{th stage error}}$ can be upper bounded as $\mathbf{E}(X_q)|_{q\text{th stage error}} \le \text{BER}_q \cdot k_q$, where BER_q denotes the decoding error probability (i.e., the block error probability) for the last stage of decoding, i.e., the block of k_q bits at the qth stage of decoding would be declared to be in error if at least one of the bits is in error. The block error probability would depend on the decoding algorithm used at the qth stage, i.e., single-stage or multistage. The block error probability can be calculated using results of [6].

A very interesting and special case of the results derived above occurs when $q = 2$ and the second level outer code is left uncoded, as shown in [1, Fig. 2]. For this special case, we can get a closed-form expression for $P_b(e)$. Using (2.1) and (2.2), $P_b(e)$ can be written in the form:

$$P_b(e) \le \left(\left(\sum_{i=1}^{2} \mathbf{E}(X_i)|_{i\text{ stage error}} \right) + \left(\mathbf{E}(X_2)|_{\text{error propagation}} \right) \cdot p_{E_2} \right) \Big/ (k_1 + k_2) \tag{2.5}$$

$\mathbf{E}(X_1)|_{1\text{st stage error}}$ can be derived using (2.4). $\mathbf{E}(X_2)|_{2\text{nd stage error}}$ can be upper bounded as

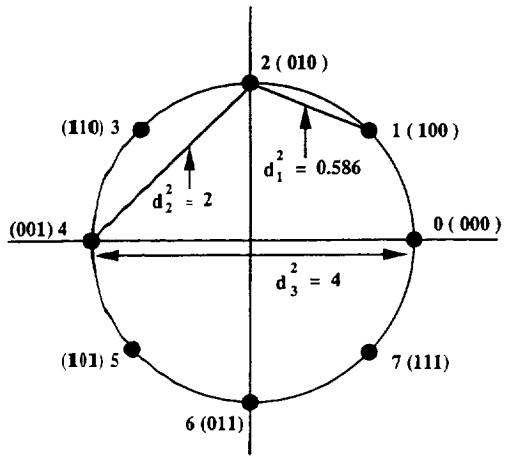

Fig. 1. An 8PSK signal constellation and its signal labels.

$\mathbf{E}(X_2)|_{\text{2nd stage error}} \leq \text{BER}_2 \cdot k_2$. Let $\mathbf{V}$ be the transmitted code sequence. Then, using [1, eq. (3.12)], $\mathbf{V}$ can be written in the form $\lambda(\phi_1(\mathbf{v_1}) + \omega_2)$, where $\mathbf{v_1}$ is a code sequence in the convolutional code used at the first level, C_1 and ω_2 is a sequence of points from Ω_2. Let the decoded code sequence associated with the convolutional code be $\mathbf{v_1} + \mathbf{e}$, where $\mathbf{e}$ is a code sequence in C_1. $p_{\mathbf{e}}$ gives us the corresponding probability of this event. Let $w_b(\mathbf{e})$ denote the branch weight of $\mathbf{e}$. Hence, the error sequence $\mathbf{e}$ will cause at most $w_b(\mathbf{e})$ blocks of k_2 bits at the second stage to be in error, i.e., the number of bits in error at the second stage of decoding, due to the error sequence $\mathbf{e}$ is $\leq k_2 \cdot w_b(\mathbf{e})$. Using arguments similar to those used to derive (2.4), $(\mathbf{E}(X_2)|_{\text{error propagation}} \cdot p_{E_2})$ can be upper bounded as $(\mathbf{E}(X_2)|_{\text{error propagation}} \cdot p_{E_2}) \leq \sum_{\mathbf{e} \in C_1} k_2 \cdot w_b(\mathbf{e}) \cdot p_{\mathbf{e}}$.

III. Examples

Examples 1 and 2 construct codes for the AWGN channel, Examples 3 and 4 construct codes for the Rayleigh-fading channel and Example 5 constructs a code for the light shadowed mobile satellite communication (MSAT) channel. In the following, we will use (n, k, d) to denote a linear block code of length n, dimension k and minimum distance d.

Example 1: Consider the case of $m = 8$, $q = 2$ and choose S = 8PSK. Hence $\ell = 3$. Fig. 1 shows the two-dimensional (2-D) 8PSK signal constellation of unit energy, in which each signal point is uniquely labeled with three bits, abc, where a is the first labeling bit and c is the last labeling bit. The labeling is done through signal partitioning process [2]. Choose $C_{0,1} = (8, 4, 4)$ Reed–Muller (RM) code, $C_{0,2} = C_{1,2} = (8, 7, 2)$ code $C_{0,3} = C_{1,3} = (8, 8, 1)$ code, and $C_{1,1} = (8, 1, 8)$ code. The minimum squared Euclidean distance of $\Lambda_0 = \lambda(\Omega_0)$ is 2.344 and for $\Lambda_1 = \lambda(\Omega_1)$ is 4.0 [6]. The encoder structure will be the same as that in [1, Fig. 2]. A rate-2/3 code will be used at the first level. Two choices will be considered for the convolutional code at the first level. The first choice is the four-state, $d_{B\text{-free}} = 2$ code from [1, Table II] and the second choice is the 16-state, $d_{B\text{-free}} = 3$ code from [1, Table II]. The phase invariance of the resulting code is the same for both the choices and is 45° and can be derived by a straightforward application of [1, Theorem 7]. The spectral efficiency is also the same for both the choices and is equal to $(16+2)/8 = 2.25$ bits/symbol. The mapping ϕ_1 used is linear. Details of ϕ_1 have been omitted due to lack of space. The following gives a detailed discussion for both the choices.

Four State: The minimum squared Euclidean distance of the code is (refer to [1, Theorem 5]): $\min\{4.0, 2.344 \cdot 2\} = 4.0$. Using [1, eq. (3.12)], any code sequence in the super trellis can be written in the form $\lambda(\phi_1(\mathbf{v_1}) + \omega_1)$, where $\mathbf{v_1}$ is code sequence in the 2/3-rate convolutional code used at the first level and ω_1 is a sequence of points from Ω_1. As such, the super trellis for this code is isomorphic to the trellis of the convolutional encoder used at the first level, with each branch of the trellis consisting of 2^{16} parallel transitions corresponding to the 2^{16} elements of Ω_1. Ω_1 has a four-state, eight-section trellis diagram [6]. Each branch of the super trellis can be expressed in the form $\lambda(\omega_0 + \Omega_1)$, where $\omega_0 \in [\Omega_0/\Omega_1]$. Hence, each branch of the super trellis has a four-state, eight-section trellis, which is *isomorphic* to the trellis of Ω_1. Standard Viterbi decoding can be used on every branch of super trellis using this four-state, eight-section *isomorphic* trellis to find the most probable parallel transition. The trellis of the overall multidimensional code can thus be viewed as a *nested trellis diagram*, i.e., a trellis within a trellis.

A reduction in the decoding complexity can be achieved by using the *multistage decoding* algorithm proposed in [1, Section V]. The decoding now proceeds in two stages. Let $\mathbf{V}$ be the transmitted code sequence. Using [1, eq. (3.12)], $\mathbf{V}$ can be written in the form $\lambda(\phi_1(\mathbf{v_1}) + \omega_1^{\text{tr}})$, where $\mathbf{v_1}$ is a code sequence in the convolutional code C_1 used at the first level, and ω_1^{tr} is a sequence of points from Ω_1. At the first stage of decoding, we form the trellis $\tilde{C}_1^{\text{sup}}$, where any code sequence in $\tilde{C}_1^{\text{sup}}$ can be written in the form $\phi_1(\mathbf{u_1}) + \omega_1^{\text{sup}}$, where ω_1^{sup} denotes a sequence of points from Ω_1^{sup} and $\mathbf{u_1}$ is a code sequence in C_1. The details of how the trellis $\tilde{C}_1^{\text{sup}}$ is formed were mentioned in [1, Section V]. Ω_1^{sup} is chosen to be: $\Omega_1^{\text{sup}} = (8, 1, 8) * (8, 8, 1) * (8, 8, 1)$ which has a very simple two-state trellis structure. On the other hand, Ω_1 has a four-state eight-section trellis diagram which is more complex than the trellis structure of Ω_1^{sup}. This helps in reducing the closest coset decoding complexity associated with the first stage of decoding. Standard Viterbi decoding is performed on the received sequence using the trellis $\lambda(\tilde{C}_1^{\text{sup}})$ to obtain an estimate of $\mathbf{v_1}$, denoted $\hat{\mathbf{v}}_1$. This completes the first stage of decoding.

At the second stage of decoding, we construct the trellis $\tilde{C}_2$, where a code sequence in $\tilde{C}_2$ is of the form $\phi_1(\hat{\mathbf{v}}_1) + \omega_1$, where ω_1 denotes a sequence of points from Ω_1. Consider the pth time instant. The structure of $\tilde{C}_2$ at the pth time instant is of the form $\tilde{C}_{2,p} = \phi_1(\hat{\mathbf{v}}_{1,p}) + \Omega_1$, where $\hat{\mathbf{v}}_{1,p}$ is the component of $\hat{\mathbf{v}}_1$ at the pth time instant. This trellis $\tilde{C}_{2,p}$ is *isomorphic* to the trellis Ω_1 and this trellis can be used to obtain an estimate of $\omega_{1,p}^{\text{tr}}$, where $\omega_{1,p}^{\text{tr}}$ is the term in ω_1^{tr} corresponding to the pth time instant.

The decoding complexity associated with the second stage of decoding can be further reduced by using the three-stage decoding technique for Ω_1 proposed by Sayegh [7] and Tanner

[8]. We will carry out the second stage of decoding using the three-stage decoding technique mentioned above.[1]

The multistage decoding algorithm does lead to a slight degradation in performance, however, as will be shown in the performance curves, the loss is negligible as compared to the reduction in complexity. The following gives the number of computations associated with both the optimal and the multistage decoding algorithm for the four-state trellis. The complexity calculation for the multistage decoding algorithm have been carried out assuming the three-stage decoding for the second stage, as mentioned above.

Computation Complexity—Optimal Decoding Algorithm: $\gamma_1 = 2$ *and* $k_1 = 2$: The branch decoding complexity B_{C_1} is: 1) since there are eight 8PSK points per branch, the distance computation complexity per branch is 64; 2) the survivor calculation for the parallel branch transitions in Ω_1 requires 32 compares; and 3) the Viterbi decoding for Ω_1 requires 52 adds and 27 comparison to calculate the final survivor (assuming the survivor for the parallel transitions has been found). Since there are eight cosets, the total complexity is 416 adds and 216 compares, i.e., B_{C_1} = 416 adds + 248 compares + 64 distance computations. Hence, total complexity is 54 adds + 32.5 compares + 8 distance computations per two dimensions.

Computation Complexity—Multistage Decoding Algorithm: $\gamma_1 = 2$ *and* $k_1 = 2$: The branch decoding complexity is:

First stage of decoding: 1) There are eight 8PSK points per branch, hence, the distance computation complexity per branch is 64; 2) the suboptimal distance estimates [8] require 48 compares; 3) Viterbi decoding of Ω_1^{sup} requires 14 adds and one compare. Since there are eight cosets, the total complexity is 112 adds and eight compares.

Second stage of decoding: 1) The multistage decoding technique requires 26 adds and 13 compares; hence, total complexity is 19.25 adds + 10.125 compares + 8 distance computations per two dimensions.

Fig. 2 shows the simulation results of the bit-error performance of both the optimal and the multistage decoding algorithm. An upper bound on the bit-error rate (BER) of the proposed code is also shown in Fig. 2. Details of the bound have been omitted due to lack of space. Also shown in the figure is the bit-error performance of a hypothetical uncoded phase shift keying (PSK) system of the same spectral efficiency [10].

Fig. 2 shows that the multistage and optimal decoding curves converge around $E_b/N_o = 8$ dB, and the performance of the optimal curve is only slightly better at low signal-to-noise ratio (SNR). The proposed code achieves a coding gain of 2.8 dB at the decoded BER of 10^{-6} over the uncoded reference system of the same spectral efficiency [10]. In addition, the decoding complexity of the optimal decoding algorithm is roughly about three times the decoding complexity of the suboptimal one.

Pietrobon *et al.* [3] do not have a comparable code over 8 × 2 dimensions, hence, comparison will be made with a 4 × 2-dimensional code over 8PSK with $\gamma = 2$ and phase invariance = 45°. Spectral efficiency of this code is 2.25 bits/symbol, same as that of the proposed code. The performance curve of this code, taken from [12], has also been shown in the figure. The complexity of the Pietrobon code is 24 adds + 17 compares + 8 distance distance computations per two dimensions. As can be seen from the figure, the proposed code outperforms the Pietrobon code by roughly 0.4 dB at $4 \cdot 10^{-6}$ BER, and in addition, the complexity of the proposed code with multistage decoding is less than that of the Pietrobon code.

16 States: The minimum squared Euclidean distance of the code is (refer to [1, Theorem 5]) $\min\{4.0, 2.344 \cdot 3\} = 4.0$. The super-trellis in this case is very similar to the four-state trellis discussed above, with the only difference that the four-state convolutional code at the first level, has been replaced by the 16-state trellis. Both the optimal and the multistage decoding techniques will be investigated for this case also. The complexity associated with the optimal and the multistage decoding technique are as follows.

Computation Complexity—Optimal Decoding Algorithm: $\gamma_1 = 4$ *and* $k_1 = 2$: The branch decoding complexity B_{C_1} is the same as the four-state case. Therefore, total complexity is 60 adds + 37 compares + 8 computations per two dimensions.

Computation-Complexity—Multistage Decoding Algorithm: $\gamma_1 = 4$ *and* $k_1 = 2$: The branch decoding complexity is the same as the four-state case. Therefore, the total complexity is 25.25 adds + 14.625 compares + 8 distance computations per two dimensions.

Fig. 3 shows the bit-error performance of the both the optimal and the suboptimal-decoding algorithm. An upper bound on the BER of the proposed code using the multistage decoding algorithm is also shown in Fig. 3.

Fig. 3 shows that the multistage and the optimal decoding curves exhibit the same characteristics as the four-state case. The two curves converge around $E_b/N_0 = 6.54$ dB, and the performance of the optimal curve is only slightly better than the optimal curve at low SNR. The proposed code achieves a coding gain of 3.2 dB at the decoded bit-error-rate of 10^{-6} over the uncoded reference system of the same spectral efficiency [10]. In addition, the decoding complexity of the optimal decoding algorithm is roughly about 2.5 times the decoding complexity of the multistage one.

Pietrobon *et al.* [3] do not have a comparable code over 8 × 2-dimensions, hence, comparison will be made with a 4 × 2-dimensional code over 8PSK with $\gamma = 3$ and phase invariance = 45°. The spectral efficiency of this code is 2.25 bits/symbol, i.e., it is the same as that of the proposed code. The performance curve of this code, taken from [9], has also been shown in the figure. The complexity of this code is 48 adds + 32 compares + 8 distance distance computations per two dimensions. The performance of the proposed code is slightly better than the Pietrobon code and in addition the complexity of the Pietrobon code is about two times higher than that of the proposed code with multistage decoding.

The 16-state proposed code with the multistage decoding algorithm achieves better performance than the four-state proposed code with the multistage decoding algorithm at the

[1] Note, the first stage of the three-stage decoding process for Ω_1 can actually be combined with the first stage of decoding of the TCM code, i.e., the stage which uses the trellis $\tilde{C}_1^{\text{sup}}$.

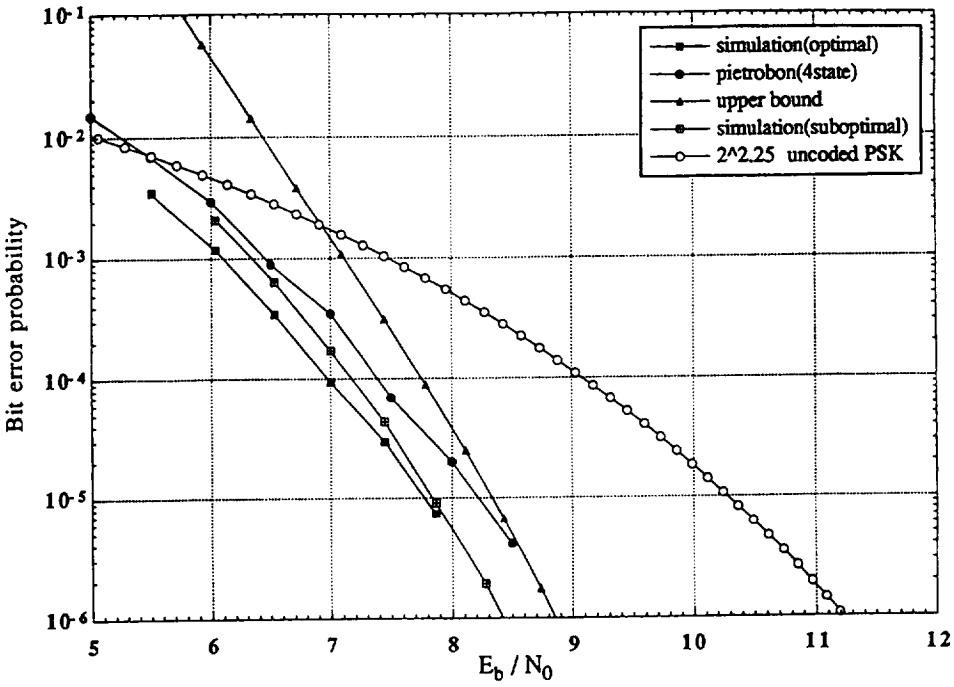

Fig. 2. BEP of the code in Example 1 with a four-state encoder at the first level for the AWGN channel.

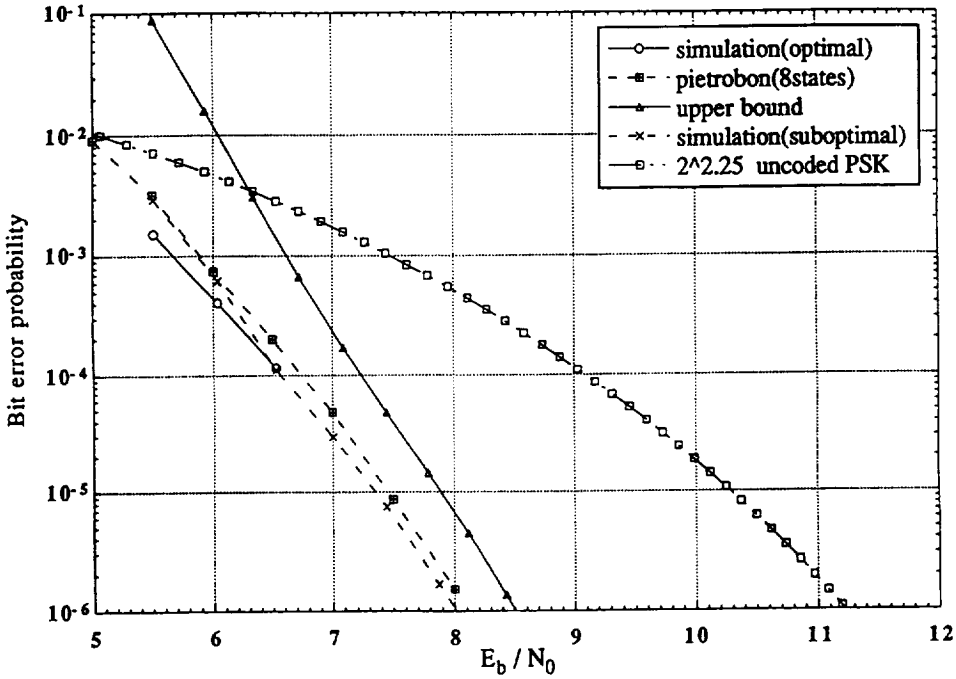

Fig. 3. BEP of the code in Example 1 with a 16-state encoder at the first level for the AWGN channel.

cost of slightly increased decoding complexity. The improvement in performance is due to the higher minimum squared Euclidean distance of the first decoding stage of the 16-state code. This leads to better performance at the first decoding stage and as a result reduced error propagation onto the second decoding stage.

Example 2: Consider the case of $m = 16$, $q = 3$ and choose $S = 8$PSK. Hence, $\ell = 3$. Choose $C_{0,1} = (16, 4, 8)$ code. This code is obtained from the first-order RM code of length 16, by removing the all ones vector from the generator matrix of the $(16, 5)$ code. Choose $C_{2,2} = (16, 11, 4)$ RM code, $C_{0,2} = C_{0,3} = C_{1,2} = C_{1,3} = C_{2,3} = (16, 15, 2)$ code and $C_{1,1} = C_{2,1} = (16, 0, \infty)$ code, i.e., the code consisting of just the all zero codeword. The minimum squared Euclidean distance for $\Lambda_0 = \lambda(\Omega_0)$ is 4.0, for $\Lambda_1 = \lambda(\Omega_1)$ is 4.0 and for $\Lambda_2 = \lambda(\Omega_2)$ is 8.0 [6]. A rate-3/4 code with 64-states (second code in [1, Table III]) will be used at the first level. Let us call this code C_1. The same rate-3/4 code used at

the first level will be used at the second level. Let us call this code C_2. The phase invariance of the resulting code is 90°. The spectral efficiency is equal to $(3+3+26)/16 = 2$ bits/symbol. The mappings ϕ_1 and ϕ_2 used at the first and second encoding levels respectively have been chosen to be linear. The minimum squared Euclidean distance of the code is at least (refer to [1, Theorem 5]), $\min\{8.0, 3{\cdot}4.0, 3{\cdot}4.0\} = 8.0$. Note that the theorem gives the minimum squared Euclidean distance associated with the first encoding stage to be at least 12.0. A quick verification of the partitions given above show that the minimum squared Euclidean distance is actually $3 \times 8 \times 0.586 = 14.064$. This is obtained by considering the squared Euclidean distance due to the (16, 4) code of Ω_0 and multiplying it by the free branch distance of C_1.

Optimal decoding of the multidimensional code would require a trellis with $2^6 \cdot 2^6 = 2^{12}$ states. Optimal decoding of the code using this 4096 state trellis would be extremely complex, and as such we will focus on the multistage decoding technique proposed in [1, Section V]. The multistage decoding of the multidimensional code proceeds in three stages.

Let $\mathbf{V}$ be the transmitted code sequence. Using [1, eq. (3.12)], $\mathbf{V}$ can be expressed in the form $\lambda(\phi_1(\mathbf{v_1}) + \phi_2(\mathbf{v_2}) + \omega_2)$, where $\mathbf{v_1}$ is a code sequence in the 64-state convolutional code C_1, $\mathbf{v_2}$ is a code sequence in the 64-state convolutional code C_2 and ω_2 is a sequence of points from Ω_2.

First stage of decoding: To simplify the trellis decoding complexity associated with the first stage of decoding, instead of forming the trellis $\tilde{C}_1$ we form the trellis $\tilde{C}_1^{\sup}$, where any code sequence in $\tilde{C}_1^{\sup}$ can be written in the form (refer to [1, Section V]), $\phi_1(\mathbf{u_1}) + \omega_1^{\sup}$, where $\omega_1^{\sup}$ is a sequence of points from $\Omega_1^{\sup}$ and $\mathbf{u_1}$ is a code sequence in C_1. $\Omega_1^{\sup}$ is chosen to be, $\Omega_1^{\sup} = (16, 0, \infty) * (16, 16, 1) * (16, 16, 1)$. $\Omega_1^{\sup}$ has a very simple one-state trellis structure. On the other hand, Ω_1 has a four-state trellis diagram which is more complex than the trellis structure of $\Omega_1^{\sup}$. This helps in reducing the closest coset decoding complexity associated with the first stage of decoding. Standard Viterbi decoding is performed on the received sequence using the trellis $\tilde{C}_1^{\sup}$ to obtain an estimate of $\mathbf{v_1}$, denoted $\hat{\mathbf{v}}_1$. This completes the first stage of decoding.

Second stage of decoding: To simplify the trellis decoding complexity associated with the second stage of decoding, instead of forming the trellis $\tilde{C}_2$, we form $\tilde{C}_2^{\sup}$, where any code sequence in $\tilde{C}_2^{\sup}$ can be written in the form (refer to [1, Section V]), $\phi_1(\hat{\mathbf{v}}_1) + \phi_2(\mathbf{u_2}) + \omega_2^{\sup}$, where $\omega_2^{\sup}$ is a sequence of points from $\Omega_2^{\sup}$ and $\mathbf{u_2}$ is a code sequence in C_2. $\Omega_2^{\sup}$ is chosen to be: $\Omega_2^{\sup} = (16, 0, \infty) * (16, 11, 4) * (16, 16, 1)$. $\Omega_2^{\sup}$ has a eight-state trellis structure [11]. On the other hand, Ω_2 has a 16-state trellis diagram which is more complex than the trellis structure of $\Omega_2^{\sup}$. This helps in reducing the closest coset decoding complexity associated with the second stage of decoding. Standard Viterbi decoding is performed on the received sequence using the trellis $\tilde{C}_2^{\sup}$ to obtain an estimate of $\mathbf{v_2}$, denoted $\hat{\mathbf{v}}_2$. This completes the second stage of decoding.

Third stage of decoding: The third stage of decoding is identical to the second stage of decoding discussed in Example 1. The three stage decoding technique proposed by Sayegh [7] and Tanner [8] is used to split up the decoding of Ω_2 into three stages. The first stage decoding of Ω_2 is trivial. Note, the second stage of the three-stage decoding process for Ω_2 can be combined with the second stage of decoding of the multidimensional TCM code.

Computation Complexity—Multistage Decoding Algorithm: $\gamma_1 = 6, k_1 = 3, \gamma_2 = 6, k_2 = 3$: The branch decoding complexity is:

First stage of decoding: 1) The distance computation complexity per branch is 128; 2) the suboptimal distance estimates require 96 compares; 3) Viterbi decoding of $\Omega_1^{\sup}$ requires three adds. Since there are 16 cosets, the total complexity is 48 adds.

Second stage of decoding: 1) The closest coset decoding for $\Omega_2^{\sup}$ requires 184 adds + 87 compares, which is the trellis decoding complexity of the (16, 11, 4) code [11]. Since there are 16 cosets, the total complexity is 2944 adds and 1392 compares.

Third stage of decoding: 1) The multistage decoding technique for Ω_2 requires 58 adds and 29 compares. Note that only the decoding complexity of the $(16, 15, 2)$ code has been taken into account. The decoding complexity of the $(16, 11, 4)$ code is included in the second stage of decoding for reasons mentioned above. Hence, the total complexity is 254.62 adds + 150.81 compares + 8 distance computations per two dimensions.

Fig. 4 shows the simulation results of the bit-error performance of multidimensional TCM code. As can be seen from the figure, the code achieves a 4.2 dB coding gain over uncoded quaternary phase shift keying (QPSK) at 10^{-6} BER. An upper bound on the BER of the proposed code using the multistage decoding algorithm is also shown in Fig. 4.

Pietrobon *et al.* [3] do not have a comparable code over 16×2 dimensions, hence comparison will be made with a 2×2-dimensional code over 8PSK with $\gamma = 7$ and phase invariance = 90°. The spectral efficiency and phase invariance of both codes is the same. This Pietrobon *et al.* code is the best in performance among all the codes listed in [3] for rate 2 bits/symbol. The performance curve of this code, taken from [9], has also been shown in the figure. The complexity of the Pietrobon code is about two times higher than that of the proposed code, however, the proposed code has performance comparable to the Pietrobon code at high SNR.

Example 3: Consider the case of $m = 2, q = 3$ and choose S = 8PSK. Hence, $\ell = 3$. Choose $C_{0,1} = C_{0,2} = C_{0,3} = C_{1,2} = C_{1,3} = (2, 2, 1)$ code, $C_{2,3} = (2, 1, 2)$ code and $C_{2,1} = C_{2,2} = C_{1,1} = (2, 0, \infty)$ code. The minimum symbol distance of $\Lambda_0 = \lambda(\Omega_0)$ is 1, for $\Lambda_1 = \lambda(\Omega_1)$ is 1 and for $\Lambda_2 = \lambda(\Omega_2)$ is 2 (refer to [1, Section III]). The other distance parameters associated with the three block modulation codes can be found by a straightforward application of the distance theorem in [5]. A rate-1/2 code with 16-states (fourth code in [1, Table I]) will be used at the first level. Let us call this code C_1. A rate-2/3 code with 16-states (second code in [1, Table II]) will be used at the second level. Let us call this code C_2. The phase invariance of the resulting code is 180°. The spectral efficiency is equal to $(1+2+1)/2 = 2$ bits/symbol. The mappings ϕ_1 and ϕ_2 have been chosen to be linear.

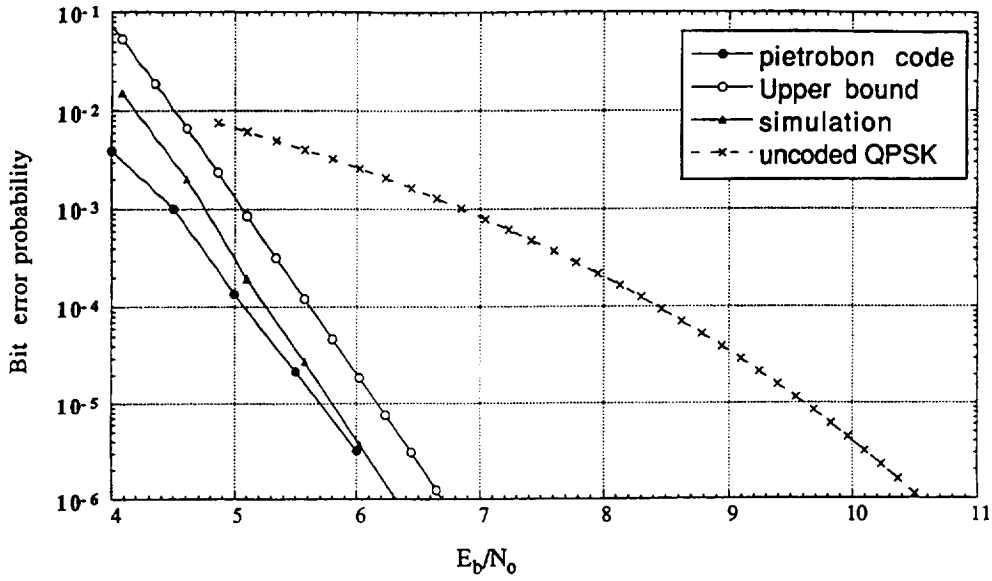

Fig. 4. BEP of the code in Example 2 for the AWGN channel.

The minimum symbol distance of the code is (refer to [1, Theorem 6]), $\min\{2, 3\cdot 1, 5\cdot 1\} = 2$. Since the minimum symbol distance of the overall modulation code is the minimum symbol distance of Λ_2, hence the minimum product distance, Δ_p^2 of the modulation code is $(4.0)^2 = 16.0$ (refer to [5]).

The decoding of this code is carried out in three stages and proceeds exactly as discussed in [1, Section V]. The second and third stage of decoding can actually be combined into one single stage of decoding. The computational complexity calculated below assumes that the second and third decoding stages have been combined.

The minimum symbol distance of the first stage is chosen to be higher than the rest of the decoding stages, so as to reduce the effect of error propagation.

Computation Complexity—Multistage Decoding Algorithm: $\gamma_1 = 4, k_1 = 1, \gamma_2 = 4, k_2 = 2$: The branch decoding complexity is:

First stage of decoding: 1) The distance computation complexity per branch is 16; 2) the suboptimal distance estimates require 12 compares; and 3) Viterbi decoding of Ω_1 requires 1 add. Since there are four cosets, the total complexity is 4 adds.

Second and third stage of decoding: 1) Viterbi decoding of Ω_2 is 2 adds + 1 compares. Since there are 8 cosets, the total complexity is 16 adds and 8 compares. Therefore, the total complexity is 58 adds + 42 compares + 8 distance computations per two dimensions.

Fig. 5 shows the simulation results of the bit-error performance of the proposed code. The performance of this code will be compared with the 16-state rate-2/3 code over 8PSK constructed by Schlegel and Costello [13] for the Rayleigh-fading channel. The spectral efficiency for both codes is the same, however, the Schlegel–Costello code has no phase invariance. The performance curve of the Schlegel–Costello code is also shown in Fig. 5. As can be seen from the figure, the proposed code outperforms the Schlegel–Costello code by about 1.6 dB at 10^{-4} bit error rate. In addition, the complexity of the Schlegel–Costello code is 64 adds + 48 compares + 8 distance computations per two dimensions which is slightly higher than that of the proposed code.

Example 4: Consider the case of $m = 8$, $q = 4$ and choose $S = 8$PSK. Hence, $\ell = 3$. Choose $C_{0,1} = C_{2,2} = C_{3,2} = C_{3,3} = (8, 4, 4)$ RM code, $C_{0,2} = C_{1,2} = (8, 7, 2)$ code, $C_{0,3} = C_{1,3} = C_{2,3} = (8, 8, 1)$ code and $C_{1,1} = C_{2,1} = C_{3,1} = (8, 0, \infty)$ code. A rate-3/4 code with eight-states (first code in [1, Table III]) will be used at the first level. Let us call this code C_1. A rate-2/3 code with 16-states (second code in [1, Table II]) will be used at the second level. Let us call this code C_2. A rate-3/4 code with 64-states (second code in [1, Table III]) will be used at the third level. Let us call this code C_3. The phase invariance of the resulting code is 180°. The spectral efficiency is equal to $(3+2+3+8)/8 = 2$ bits/symbol. The mappings ϕ_1, ϕ_2, ϕ_3 and ϕ_4 are chosen to be linear.

The decoding of this code is carried out in four stages and proceeds in a manner similar to that in Example 2. The first stage of decoding is similar to the first stage of decoding in Example 2. $\Omega_1^{\sup}$ used to simplify the decoding complexity is:$\Omega_1^{\sup} = (8, 0, \infty) * (8, 8, 1) * (8, 8, 1)$. $\Omega_1^{\sup}$ has a very simple one-state trellis which is less complex than the two-state trellis of Ω_1. The second and third stage of decoding is carried out exactly as described in [1, Section V]. The fourth stage of decoding is carried out using the multistage decoding technique for Ω_3 (as was explained in Example 1). The multistage decoding of Ω_3 proceeds in two stages. The first stage of decoding decodes the code $C_{3,2}$ and the second stage decodes the $C_{3,3}$ code. The decoding of $C_{3,2}$ can be merged with the second stage of decoding of the proposed code, and the decoding of $C_{3,3}$ can be merged with the third stage decoding of the proposed code. The complexity calculations given below assume that the fourth stage of decoding of the proposed code has been merged with the previous stages.

Computation Complexity—Multistage Decoding Algorithm: $\gamma_1 = 3, k_1 = 3, \gamma_2 = 4, k_2 = 2, \gamma_3 = 6, k_3 = 3$: The branch decoding complexity is:

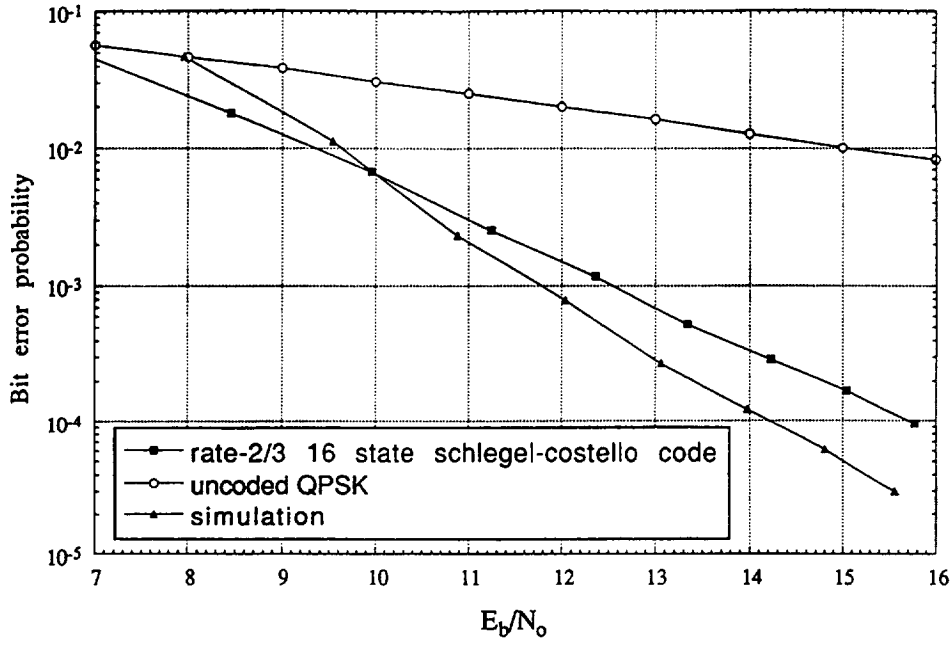

Fig. 5. BEP of the code in Example 3 for the Rayleigh-fading channel.

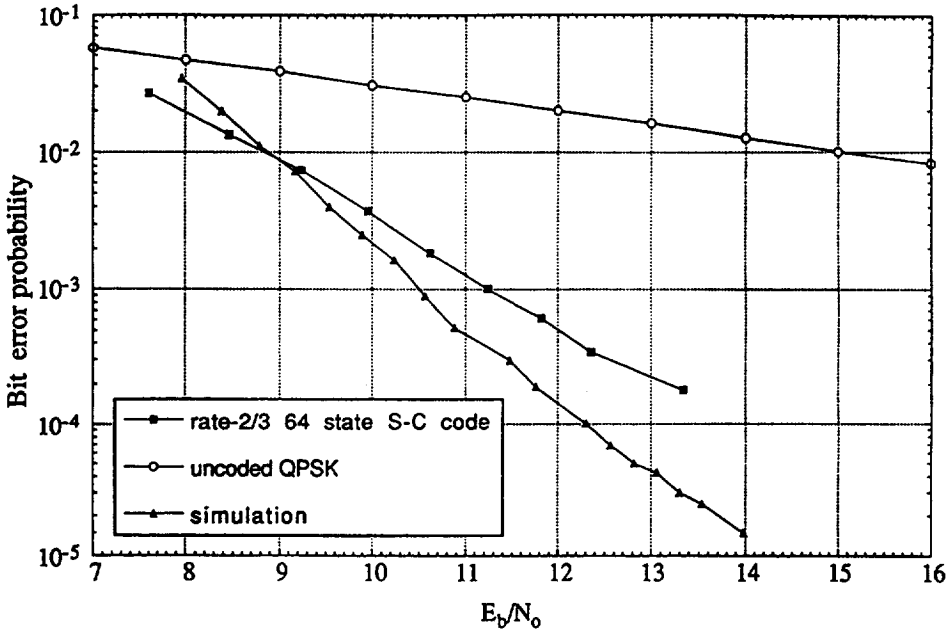

Fig. 6. BEP of the code in Example 4 for the Rayleigh-fading channel.

First stage of decoding: 1) The distance computation complexity per branch is 64; 2) the suboptimal distance estimates require 48 compares; 3) Viterbi decoding of $\Omega_1^{\sup}$ requires 7 adds. Since there are 16 cosets, the total complexity is 112 adds.

Second stage of decoding and the first stage of the fourth stage of decoding: 1) The closest coset decoding complexity is 36 adds and 11 compares, which is the trellis decoding complexity of the (8, 4, 4) code [11]. Since there are 8 cosets, the total complexity is 288 adds and 88 compares.

Third stage of decoding and the second stage of the fourth stage of decoding: 1) The closest coset decoding complexity is 36 adds and 11 compares, which is the trellis decoding complexity of the (8, 4, 4) code [11]. Since there are 16 cosets, the total complexity is 576 adds and 176 compares. Therefore, total complexity is 202 adds + 108 compares + 8 distance computations per two dimensions.

Fig. 6 shows the simulation results of the bit-error performance of the proposed code. The performance of this code will be compared with the 64-state rate-2/3 code over 8PSK constructed by Schlegel and Costello [13] for the Rayleigh-fading channel. The spectral efficiency for both codes is the same, however the Schlegel–Costello code has no phase invariance. The performance curve of the Schlegel–Costello

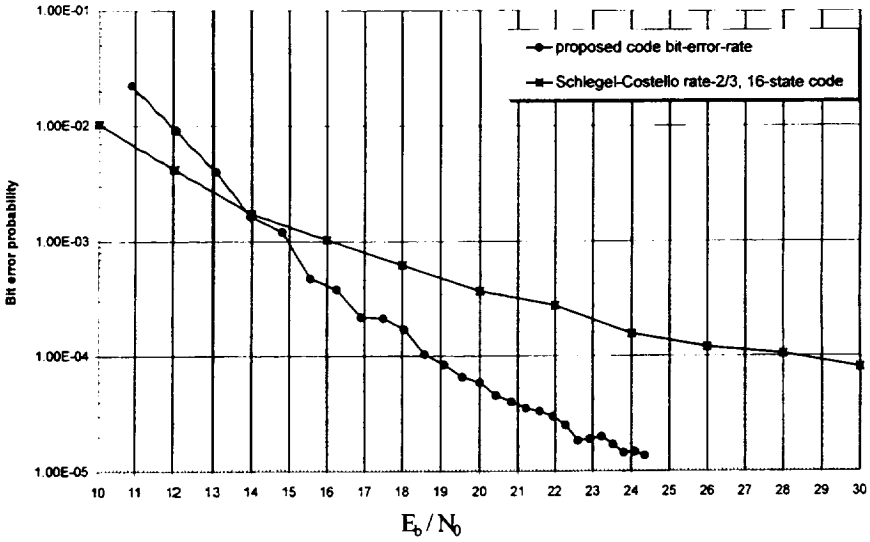

Fig. 7. BEP of the code in Example 5 for ashadowed mobile satellite channel.

code is also shown in Fig. 6. As can be seen from the figure, the proposed code outperforms the Schlegel–Costello code by about 1.5 dB at $2 \cdot 10^{-4}$ BER. In addition, the complexity of the Schlegel–Costello code is 256 adds + 192 compares + 8 distance computations per two dimensions which is higher than that of the proposed code.

Example 5: A statistical model for the shadowed mobile satellite channel has been devised by Loo [14]–[17] and this model has been used by other researchers [18]–[23] to study the error performance of coded modulation schemes over the MSAT channel. In Loo's model, there are three different kinds of shadowing-light, average and heavy. The corresponding Rician factors are 6.16, 5.46, and −19.33 dB, respectively. Therefore, in the shadowed MSAT channel, a coded modulation system suffers very severe distortion due to randomly changing phase and multipath fading. Especially, if the Doppler frequency shift is large due to the motion of vehicle, a coded modulation system faces the error floor phenomenon. We will assume that the carrier frequency is 870 MHz and the symbol rate is 2400 symbols/s. Due to randomly changing phase, perfect phase synchronization is not feasible in the shadowed MSAT channel. Therefore, differentially detected 8PSK modulation is used. We assume that the speed of moving object is 92.88 miles/h. The corresponding normalized fading bandwidth BT is 0.05, where B is the maximum Doppler frequency shift and T^{-1} is the symbol rate. To combat burst errors, a block interleaver is used for computer simulation. The size of interleaver is 512 8DPSK symbols, and the number of rows of the block interleaver is 64 and the number of columns is 8.

Consider the case of $m = 8$ and $q = 3$. Hence, $\ell = 3$. Choose $C_{0,1} = C_{2,2} = (8,4,4)$ RM code, $C_{0,2} = C_{0,3} = C_{1,2} = C_{1,3} = C_{2,3} = (8,7,2)$ code and $C_{1,1} = C_{2,1} = (8,0,\infty)$ code. A rate-3/4 code with eight-states (first code in [1, Table III]) will be used at the first level. Let us call this code C_1. A rate-2/3 code with 16-states (second code in [1, Table II]) will be used at the second level. Let us call this code C_2. The phase invariance of the resulting code is 90°. The spectral efficiency is equal to $(3 + 2 + 11)/8 = 2$ bits/symbol. The mappings ϕ_1 and ϕ_2 used at the first and second encoding levels are linear.

Decoding of the code proceeds exactly as in Example 2, and as such, will not be repeated here. The complexity calculations are also very similar to Example 2, and as such details will be omitted. The total complexity is 69.25 adds + 31.63 compares + 8 distance computations per two dimensions.

Fig. 7 shows the simulation results of the bit-error performance of the proposed code. The performance of this code will be compared with the 16-state rate-2/3 code constructed by Schlegel and Costello [13] (this code is chosen, for lack of comparable complexity code available in literature for the shadowed MSAT channel). The spectral efficiency for both codes is the same. The performance curve of the Schlegel–Costello code is also shown in Fig. 7. As can be seen from the figure, the proposed code outperforms the Schlegel–Costello code by about 9.65 dB at 10^{-4} bit error rate. Also, the proposed code faces the error floor at around 1.4×10^{-5} BER, whereas the Schlegel–Costello code faces an error floor around 4.8×10^{-5} BER. In addition, the complexity of the Schlegel–Costello code is higher than that of the proposed code.

IV. Conclusion

A simple and systematic technique of constructing multidimensional TCM codes using block modulation codes and convolutional codes optimized for branch distance is proposed. Bounds on the minimum squared Euclidean distance and minimum symbol distance of the multidimensional TCM codes are derived, along with conditions on phase invariance. A multistage decoding technique for the multidimensional TCM codes has also been proposed. Examples constructed show that the technique can be used to construct good codes which have a performance/decoding complexity advantage over the codes available in literature for both the AWGN and fading channels.

REFERENCES

[1] S. Rajpal, D. J. Rhee, and S. Lin, " Multidimensional trellis coded phase modulation using a multilevel concatenation approach—Part I: Code design," *IEEE Trans. Commun.*, vol. 45, pp. 64–72, Jan. 1997.

[2] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 55–67, Jan. 1982.

[3] S. S. Pietrobon *et al.*, "Trellis coded multidimensional phase modulation," *IEEE Trans. Inform. Theory*, vol. 36, pp. 63–89, Jan. 1990.

[4] D. Divsalar and M. K. Simon, "The design of trellis coded MPSK for fading channels: Performance criteria," *IEEE Trans. Commun.*, pp. 1004–1012, Sept. 1988.

[5] J. Wu and S. Lin, "Multilevel trellis MPSK modulation codes for the rayleigh fading channel," *IEEE Trans. Commun.*, vol. 41, Sept. 1993.

[6] T. Kasami *et al.*, "On multilevel block modulation codes," *IEEE Trans. Inform. Theory*, vol. 37, no. 4, pp. 965–975, July 1991.

[7] S. I. Sayegh, "A class of optimum block codes in signal space," *IEEE Trans. Commun.*, vol. COM-30, pp. 1043–1045, Oct. 1986.

[8] R. M. Tanner, "Algebraic construction of large euclidean distance combined coding modulation systems," in *Abstract of Papers, 1986 IEEE Int. Symp. Inform. Theory*, Ann Harbor, Oct. 6–9, 1986.

[9] D. J. Costello Jr., private communication.

[10] R. H. Deng and D. J. Costello Jr., "High rate concatenated coding systems using multidimensional bandwidth efficient trellis inner codes," *IEEE Trans. Commun.*, vol. 37, pp. 1091–1096, Oct. 1989.

[11] G. D. Forney Jr., "Coset codes—Part II: Binary lattices and related codes," *IEEE Trans. Inform. Theory*, vol. IT-34, pp. 1152–1187, Sept. 1988.

[12] L. C. Perez and D. J. Costello Jr., "On the performance of multidimensional phase modulated trellis codes," NASA Tech. Rep. 89-10-02, NASA Training Grant NGT-7010, Oct. 1989.

[13] C. Schlegel and D. J. Costello Jr., "Bandwidth efficient coding for fading channels: Code construction and performance analysis," *IEEE J. Select. Areas Commun.*, vol. 7, no. 9, Dec. 1989.

[14] C. Loo, "Measurements and models of a mobile-satellite link with applications," in *Proc. GLOBECOM'85*, Dec. 2–5, 1985.

[15] ——, "A statistical model for a land mobile satellite link," in *Links, for the Future (ICC'84), Science, Systems, and Services for Comm.*, P. Dewilde and C. A. May, Eds. New York: North-Holland, 1984.

[16] ——, "A statistical model for a land mobile satellite link," *IEEE Trans. Veh. Technol.*, vol. VT-34, pp. 122–127, Aug. 1985.

[17] C. Loo *et al.*, "Measurements and modeling of land-mobile satellite signal statistics," presented at the IEEE 1986 Vehicular Technology Conf., May 20–22, 1986.

[18] P. J. McLane *et al.*, "PSK and DPSK trellis codes for fast fading, shadowed mobile satellite communication channels," in *Proc. 1987 Int. Conf. Commun.*, June 7–10, 1987, pp. 21.1.1–21.1.6.

[19] ——, "PSK and DPSK trellis codes for fast fading, shadowed mobile satellite communication channels," *IEEE Trans. Commun.*, vol. 36, pp. 1242–1246, Nov. 1988.

[20] A. C. M. Lee and P. McLane, "Conventionally interleaved PSK and DPSK trellis codes for shadow, fast fading, mobile satellite communications channel," *IEEE Trans. Veh. Technol.*, vol. 39, pp. 37–47, Feb. 1990.

[21] R. G. McKay *et al.*, "Analytical performance bounds on average bit error probability for trellis coded PSK transmitted over fading channels," in *Proc. IEEE Int. Conf. Commun.*, 1989, pp. 9.2.1–9.2.7.

[22] S. H. Jamali and T. Le-Ngoc, "A new 4-state 8PSK TCM scheme for fast fading, shadowed mobile radio channels," *IEEE Trans. Veh. Technol.*, pp. 216–222, Feb. 1991.

[23] ——, "Performance comparison of different decoding strategies for a bandwidth-efficient block-coded scheme on mobile radio channels," *IEEE Trans. Veh. Technol.*, vol. 41, no. 4, pp. 505–515, Nov. 1993.

Sandeep Rajpal (S'88-94 M'95) for a photograph and biography, see p. 72 of the January 1997 issue of this TRANSACTIONS.

Do Jun Rhee (S'85–M'96) for a photograph and biography, see p. 72 of the January 1997 issue of this TRANSACTIONS.

Shu Lin (F'80) for a photograph and biography, see p. 63 of the January 1997 issue of this TRANSACTIONS.